

## 定时器 T1-查询方式

### 1. 实验目的

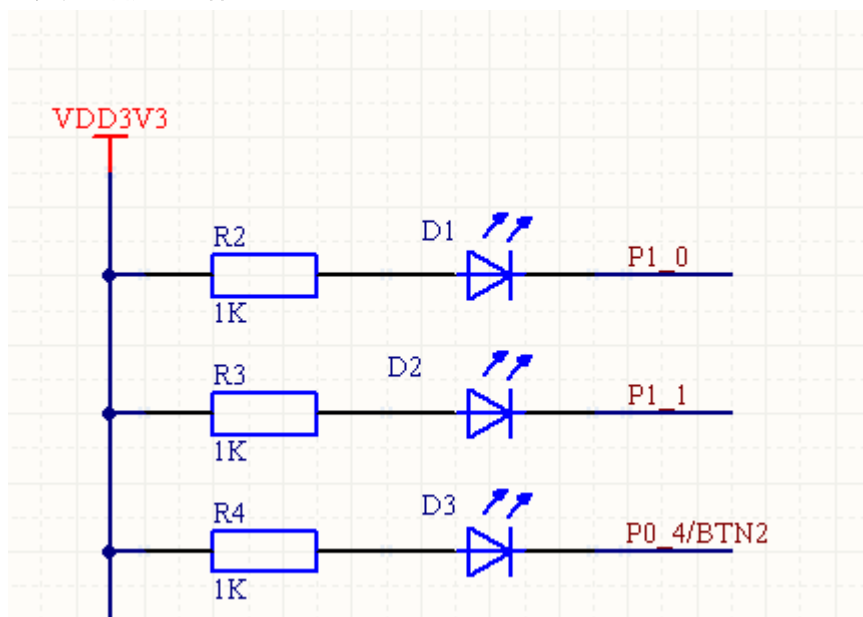
- 1)、通过实验掌握 CC2530 芯片 GPIO 的配置方法
- 2)、掌握 Led 驱动电路及开关 Led 的原理
- 3)、掌握定时器 T1 的配置与使用

### 2. 实验设备

硬件：PC 机一台 ZB 网关（底板、核心板、仿真器、USB 线）一套

软件：2000/XP/win7 系统，IAR 8.10 集成开发环境

### 3. 实验相关电路图



发光二极管是属于二极管的一种，具有二极管单向导电特性，即只有在正向电压（二极管的正极接正，负极接负）下才能导通发光。P1.0 引脚接发光二极管(D1)的负极,所以 P1.0 引脚输出低电平 D1 亮，P1.0 引脚输出高电平 D1 熄灭,D2,D3 同理。

#### 4. 实验相关寄存器

CC2530 的 T1 定时器（16 位）需要配置三个寄存器 T1CTL、T1STAT、IRCON。如下表所示：

寄存器	作用	描述
T1CTL(0xE4)	定时器 1 的控制 和状态	T1CTL (Bit 3:2) 分频器划分值，如下： 00： 标记频率/1 01： 标记频率/8 10： 标记频率/32 11： 标记频率/128 T1CTL (Bit 1:0) 选择定时器 1 模式 00： 暂停运行。 01： 自由运行，从 0x0000 到 0xFFFF 反复计数。 10： 模，从 0x0000 到 T1CC0 反复计数。 11： 正计数/倒计数，从 0x0000 到 T1CC0 反复计数并 且从 T1CC0 倒计数到 0x0000。
T1STAT(0xAF)	定时器 1 状态	Bit5：定时器 计数器溢出中断标志 Bit4：定时器 1 通道 4 中断标志 Bit3：定时器 1 通道 3 中断标志 Bit2：定时器 1 通道 2 中断标志 Bit1：定时器 1 通道 1 中断标志 Bit0：定时器 1 通道 0 中断标志
IRCON(0xC0)	中断标志 4	Bit1：定时器 1 中断标志。当定时器 1 中断发生时设为 1 并且当 CPU 向量指向中断服务例程时清除。 0： 无中断未决 1： 中断未决

按照表格寄存器的内容，对 Led1 和 T1 进行配置。

Led1、LED2、LED3 配置如下：

```
P1DIR = 0x03;          //P10 P11 为输出
```

```
P1DIR = 0x10;           //P14 为输出
```

T1 配置如下：

```
T1CTL = 0x05;
```

```
T1STAT= 0x21;           //通道 0,中断有效,8 分频;自动重装模式(0x0000->0xffff) 5.源码分析
```

## 5.源码分析

```
#include <ioCC2530.h>
```

```
#define uint  unsigned int
```

```
#define uchar  unsigned char
```

```
#define LED1  P1_0           //定义 LED1 为 P10 口控制
```

```
#define LED2  P1_1           //定义 LED2 为 P11 口控制
```

```
#define LED3  P0_4           //定义 LED3 为 P04 口控制
```

```
uint counter=0;             //统计溢出次数
```

```
uint LEDFlag;               //标志是否要闪烁
```

```
void InitialT1test(void);    //初始化函数声明
```

```
/******
```

```
//T1 初始化程序
```

```
******/
```

```
void InitialT1test(void)
```

```
{
```

```
    //初始化 LED 控制端口 P1
```

```
    P1DIR = 0x03;           //P10 P11 为输出
```

```
    P0DIR = 0x10;           //P04 为输出
```

```
    //灯默认为灭
```

```
LED1 = 1;
LED2 = 1;
LED3 = 1;

//初始化计数器 1
T1CTL = 0x05;
T1STAT= 0x21;    //通道 0,中断有效,8 分频;自动重装模式(0x0000->0xffff)
}
/*****
//主函数
*****/
void main()
{
    InitialT1test();    //调用初始化函数
    while(1)           //查询溢出
    {
        if(IRCON > 0)
        {
            IRCON = 0;    //清溢出标志
            counter++;

            if(counter==15)    //中断计数, 约 0.25s
            {
                counter =0;
                LEDFlag = !LEDFlag;
            }
        }

        if(LEDFlag)
        {
            LED2 = LED1;
        }
    }
}
```

```

LED3 = !LED2;
LED1 = !LED1;  // 每 1s LED 灯闪烁一下
LEDFlag = !LEDFlag; // 闪烁标志变量置 0
}
}
}

```

## 6.实验现象

当把程序下到开发板后，会看到 LED1\LED3 和 LED2 交替闪烁。\_\_

